Submitted to the Bernoulli
arXiv: math.PR/0000000

ϵ-Strong Simulation of the Brownian Path

ALEXANDROS BESKOS[1] STEFANO PELUCHETTI[2] and GARETH ROBERTS[3]

[1] *Department of Statistical Science, UCL, Gower Street, London, WC1E 6BT, UK*
E-mail: alex@stats.ucl.ac.uk

[2] *HSBC Bank, 8 Canada Square, London, E14 5HQ, UK*
E-mail: stefano.peluchetti@hsbcib.com

[3] *Department of Statistics, University of Warwick, Coventry, CV4 7AL, UK*
E-mail: gareth.o.roberts@warwick.ac.uk

We present an iterative sampling method which delivers upper and lower bounding processes for the Brownian path. We develop such processes with particular emphasis on being able to unbiasedly simulate them on a personal computer. The dominating processes converge almost surely in the supremum and L_1 norms. In particular, the rate of converge in L_1 is of the order $\mathcal{O}(\mathcal{K}^{-1/2})$, $\mathcal{K}$ denoting the computing cost. The a.s. enfolding of the Brownian path can be exploited in Monte Carlo applications involving Brownian paths whence our algorithm (termed the ϵ-strong algorithm) can deliver *unbiased* Monte-Carlo estimators over path expectations, overcoming discretisation errors characterising standard approaches. We will show analytical results from applications of the ϵ-strong algorithm for estimating expectations arising in option pricing. We will also illustrate that individual steps of the algorithm can be of separate interest, giving new simulation methods for interesting Brownian distributions.

Keywords: Brownian bridge, unbiased sampling, intersection layer, pathwise convergence, iterative algorithm, option pricing.

1. Introduction

Brownian motion (BM) is an object of paramount significance in stochastic modelling. Starting from its original mathematical formulation by [2], its properties are still under meticulous investigation by contemporary researchers. Relevant to the purposes of this paper, considerable work has focused on various constructions and representations of BM paths. Leaving aside the simple finite-dimensional Gaussian structure of BM, researchers have often been interested on more complex functionals. Hitting times, extremes, local times, reflections and other characteristics of BM have been investigated (for a general exposition see [18]). For simulation purposes, many of the relevant distributions are easy to sample from on a computer ([10]). Several *conditioned* constructions of BM are also known relating BM with the Bessel process, the Rayleigh distribution and other stochastic objects (see e.g. [3]).

This paper presents a contribution of our own at simulation methods for Brownian dynamics. We develop an iterative sampling algorithm, the ϵ-strong algorithm, which simulates upper and lower paths enveloping a.s. the Brownian path. To meet this objective, we collect a number of characterisations and combine them in a way that they can deliver simple sampling methods implementable on a personal computer. We will show that after $\mathcal{O}(\mathcal{K})$-computational effort, the dominating process have L_1-distance of $\mathcal{O}(\mathcal{K}^{-1/2})$. This a.s. enfolding of the Brownian path can be exploited in Monte Carlo applications involving Brownian motion integrals, minima, maxima or hitting times; in such scenaria, the ϵ-strong algorithm can deliver unbiased Monte-Carlo estimators over Brownian expectations, overcoming discretization errors characterising standard approaches (for the latter approaches, see for instance the exposition in [12] in the context of applications in finance).

We will show applications of the algorithm and experimentally compare the required computing resources against typical alternatives employed in the literature involving Euler approximation. Our examples will involve a collection of double-barrier option pricing problems in a Black and Scholes framework arising in finance. Also, we will demonstrate that individual steps of the algorithm can be of separate interest, giving new simulation methods for interesting Brownian distributions.

The ϵ-strong algorithm delivers a pair of dominating processes, denoted by $X^{\downarrow}(n) = \{X^{\downarrow}_u(n); u \in [0,1]\}$ and $X^{\uparrow}(n) = \{X^{\uparrow}_u(n); u \in [0,1]\}$, that can be simulated on a personal computer without any discretisation error, with the property:

$$X^{\downarrow}_u(n) \leq X^{\downarrow}_u(n+1) \leq X_u \leq X^{\uparrow}_u(n+1) \leq X^{\uparrow}_u(n) \tag{1.1}$$

for all instances $u \in [0,1]$; here, X is the Brownian path. The two dominating processes will converge in the limit:

$$\text{w.p.1}\ , \quad \lim_{n\to\infty} \sup_{u\in[0,1]} \left| X^{\uparrow}_u(n) - X^{\downarrow}_u(n) \right| \to 0\ . \tag{1.2}$$

The algorithm builds on the notion of the *intersection layer*, a collective information, containing the starting and ending points of a Brownian path together with information about its extrema. A number of operations (bisection, refinement, see main text) can be applied on this information, explicitly on a computer, allowing the sampler to iterate itself to get closer to X.

We should note here that the methods described in this paper will be relevant also for non-linear Stochastic Differential Equations (SDEs). Recent developments in the simulation of SDEs under the framework of the so-called ‘Exact Algorithm’ (see [8, 7, 5, 13, 6, 9]) build upon the result that, conditionally on a collection of randomly sampled points, the path of the SDE is made of independent Brownian paths. Once this collection of points is sampled, the methodology of this paper can then be applied separately on each of the constituent Brownian sub-paths.

The structure of the paper is as following. In Section 2 we present the notion of the *intersection layer* which will be critical for our methods. In Section 3 we present the individual steps forming the ϵ-strong algorithm; they will require original simulation techniques for some Brownian distributions. Once we identify in Section 4 the ζ*-function*, an alternating monotone series at the core of Brownian dynamics, we exploit its structure in Section 5 to analytically develop these new sampling methods. In Section 6 we apply the

ϵ-strong algorithm to unbiasedly estimate some path expectations arising when pricing options in finance. We will contrast the computational cost of the algorithm with Euler approximation alternatives to get a better understanding of its practical competitiveness. In Section 7 we sketch some other potential applications of the ϵ-strong algorithm. We finish with some discussion and conclusions in Section 8.

2. Intersection Layer and Operations

We will, in general, write paths as $X = \{X_u; u \in [s,t]\}$ for $s < t$. A Brownian bridge on $[s,t]$ is a Brownian motion conditioned to start at X_s and end at X_t, for some pre-specified X_s, X_t; its finite-dimensional dynamics are easily derivable following this interpretation (see for instance [18]).

Instrumental in our considerations is the notion of (what we call) the *intersection layer*. Consider a Brownian bridge X on $[s,t]$. Let $m_{s,t}$, $M_{s,t}$ be the extrema of X:

$$m_{s,t} = \inf\{X_u; u \in [s,t]\} \ , \quad M_{s,t} = \sup\{X_u; u \in [s,t]\} \ .$$

The ϵ-strong algorithm will require some information on both $m_{s,t}$ and $M_{s,t}$. We will identify intervals:

$$\mathcal{U}_{s,t} = [U^{\downarrow}_{s,t}, U^{\uparrow}_{s,t}] \ , \quad \mathcal{L}_{s,t} = [L^{\downarrow}_{s,t}, L^{\uparrow}_{s,t}] \ ,$$

such that:

$$M_{s,t} \in \mathcal{U}_{s,t} \ , \quad m_{s,t} \in \mathcal{L}_{s,t} \ .$$

We will write simply m, M, $U^{\uparrow}$, $U^{\downarrow}$, $L^{\uparrow}$, $L^{\downarrow}$ ignoring the s,t-subscripted versions when the time interval under consideration is clearly implied by the context. The intersection layer idea refers to the collective information

$$\mathcal{I}_{s,t} = \{X_s, X_t, \mathcal{L}_{s,t}, \mathcal{U}_{s,t}\} \ , \tag{2.1}$$

that is the starting and ending points of the bridge together with intervals that contain its maximum and minimum. Fig.1(a) presents a graphical illustration of the intersection layer: the extrema of an underlying Brownian bridge lie in the shaded rectangles. We will look now at two simple operations on the information $\mathcal{I}_{s,t}$ which nonetheless will be the building blocks of the complete ϵ-strong algorithm described in the next section.

2.1. Refining the Information $\mathcal{I}_{s,t}$

During the iterations at the execution of the ϵ-strong algorithm, for each piece of information $\mathcal{I}_{s,t}$ we will need to control the width of the layers $\mathcal{L}_{s,t}, \mathcal{U}_{s,t}$ relatively to the size $t-s$ of the time interval to ensure convergence of the bounding paths enveloping the underlying Brownian path, Thus, the refinement of the information $\mathcal{I}_{s,t}$ corresponds to a procedure that updates $\mathcal{I}_{s,t}$ by halving the allowed width for the minimum m or the maximum M of the path, thereby correspondingly updating the layers $\mathcal{L}_{s,t}$ or $\mathcal{U}_{s,t}$.

```
Bisect($\mathcal{I}_{s,t}$):
  1.  Set $t^* = (t+s)/2$. Simulate $X_{t^*}$ given $\mathcal{I}_{s,t}$. Set $U^\downarrow = U^\downarrow \vee X_{t^*}$, $L^\uparrow = L^\uparrow \wedge X_{t^*}$.
  2a. Decide if $\mathcal{U}_{s,t^*} = [X_s \vee X_{t^*}, U^\downarrow]$ or $[U^\downarrow, U^\uparrow]$.
  2b. Decide if $\mathcal{U}_{t^*,t} = [X_{t^*} \vee X_t, U^\downarrow]$ or $[U^\downarrow, U^\uparrow]$.
  2c. Decide if $\mathcal{L}_{s,t^*} = [L^\downarrow, L^\uparrow]$ or $[L^\uparrow, X_s \wedge X_{t^*}]$.
  2d. Decide if $\mathcal{L}_{t^*,t} = [L^\downarrow, L^\uparrow]$ or $[L^\uparrow, X_{t^*} \wedge X_t]$.
  3.  Return $\mathcal{I}_{s,t^*} \bigvee \mathcal{I}_{t^*,t}$.
```

Table 1. The procedure for bisecting the information $\mathcal{I}_{s,t}$. It returns the intersection layers $\mathcal{I}_{s,t^*}$ and $\mathcal{I}_{t^*,t}$ with refined information about the underlying path (compared to $\mathcal{I}_{s,t}$).

More analytically, refinement of $\mathcal{I}_{s,t}$ corresponds to deciding whether the minimum m on $[s,t]$, already known to be in $[L^\downarrow, L^\uparrow]$, lies in $[L^\downarrow, (L^\downarrow + L^\uparrow)/2]$ or $[(L^\downarrow + L^\uparrow)/2, L^\uparrow]$, i.e. whether $\mathcal{L}_{s,t}$ is equal to $[L^\downarrow, (L^\downarrow + L^\uparrow)/2]$ or $[(L^\downarrow + L^\uparrow)/2, L^\uparrow]$; the apparent analogue of such a consideration applies for the maximum M. The analytical method of sampling the relevant binary random variables for carrying out this procedure will be described in Section 5.

2.2. Bisecting the Information $\mathcal{I}_{s,t}$

This is a more involved operation on $\mathcal{I}_{s,t}$, and involves bisecting $\mathcal{I}_{s,t}$ into the more analytical information $\mathcal{I}_{s,t^*} \vee \mathcal{I}_{t^*,t}$ for some intermediate time instance $t^* \in (t,s)$. In particular, we will be selecting $t^* = (t+s)/2$ within the ϵ-strong algorithm. The method begins by sampling the middle point X_{t^*} conditionally on $\mathcal{I}_{s,t}$, and then appropriately sampling the layers for the two pieces of information $\mathcal{I}_{s,t^*}$, $\mathcal{I}_{t^*,t}$. The practicalities of implementing the second part of the method will depend on whether X_{t^*} falls within a layer of $\mathcal{I}_{s,t}$ or not, thus we present the bisection operation in more detail in Table 1.

Note that if $X_{t^*} > U^\downarrow$ the two upper layers (for $\mathcal{I}_{s,t^*}$ and $\mathcal{I}_{t^*,t}$) will be directly set to $[X_{t^*}, U^\uparrow]$, and we will have to simulate extra randomness about the underlying path only to determine the lower layers. Correspondingly, if $X_{t^*} < L^\uparrow$ the two lower layers will immediately be set to $[L^\downarrow, X_{t^*}]$. In the scenario when $L^\downarrow < X_{t^*} < U^\downarrow$ we will have to simulate extra randomness to determine all four layers. We describe in Section 5 the algorithms for sampling X_{t^*} and determining the layers. Fig.1 shows a graphical illustration of the bisection procedure.

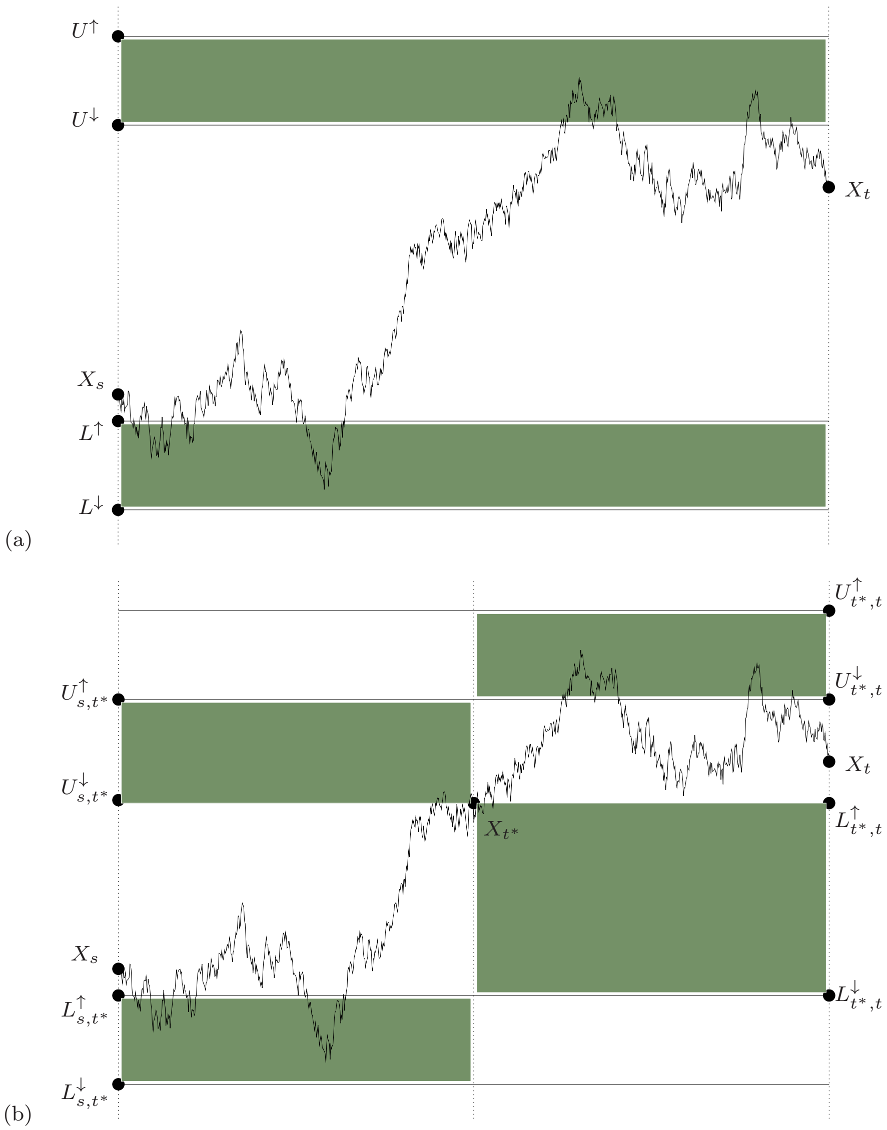


Figure 1. Top panel (a): the intersection layer information $\mathcal{I}_{s,t}$ for a Brownian path. The underlying trajectory starts at X_s and finishes at X_t with its extrema found in the shaded areas. Bottom panel (b): the bisection of $\mathcal{I}_{s,t}$ into $\mathcal{I}_{s,t^*}$ and $\mathcal{I}_{t^*,t}$. The algorithm simulates X_{t^*} and then decides that the extrema for each of the intervals $[s,t^*]$ and $[t^*,t]$ are in the shaded areas, i.e. $\mathcal{U}_{s,t^*} = [X_{t^*}, U^{\downarrow}]$, $\mathcal{U}_{t^*,t} = [U^{\downarrow}, U^{\uparrow}]$, $\mathcal{L}_{s,t^*} = [L^{\downarrow}, L^{\uparrow}]$ and $\mathcal{L}_{t^*,t} = [L^{\uparrow}, X_{t^*}]$. The algorithm outputs the upgraded information $\mathcal{I}_{s,t^*} = \{X_s, X_{t^*}, \mathcal{L}_{s,t^*}, \mathcal{U}_{s,t^*}\}$ and $\mathcal{I}_{t^*,t} = \{X_{t^*}, X_t, \mathcal{L}_{t^*,t}, \mathcal{U}_{t^*,t}\}$.

```
$\epsilon$-strong($X_0, X_1, n$):

  1. Initialize $\mathcal{U}_{0,1}$, $\mathcal{L}_{0,1}$, set $\mathcal{I}_{0,1} = \{X_0, X_1, \mathcal{U}_{0,1}, \mathcal{L}_{0,1}\}$. Set $\mathcal{P} = \{\mathcal{I}_{0,1}\}$ and $i = 1$.
  2. For each of the $2^{i-1}$ intersection layers in $\mathcal{P}$, say $\mathcal{I}_{s,t}$, do the following:
       i. Bisect the information $\mathcal{I}_{s,t}$ into $\mathcal{I}_{s,t^*}$, $\mathcal{I}_{t^*,t}$, where $t^* = (t+s)/2$.
      ii. Refine $\mathcal{I}_{s,t^*}$, $\mathcal{I}_{t^*,t}$ until the width of their layers is not greater than $\sqrt{(t-s)/2}$.
  3. Collect the updated information, $\mathcal{P} = \bigvee_{j=1}^{2^i} \mathcal{I}_{(j-1)2^{-i},j2^{-i}}$.
  4. If $i < n$ set $i = i + 1$ and return to Step 2; otherwise return $\mathcal{P}$.
```

Table 2. The ϵ-strong algorithm. It iteratively unveils extra information about the underlying path. It outputs the collection of intersection layers $\mathcal{P} = \bigvee_{j=1}^{2^n} \mathcal{I}_{(j-1)2^{-n},j2^{-n}}$.

3. ϵ-Strong Simulation of Brownian Path

We introduce an iterative simulation algorithm with input a Brownian bridge X on the domain $[0,1]$ and output, after n iterations, upper and lower dominating processes $X^{\downarrow}(n) = \{X_u^{\downarrow}(n); u \in [0,1]\}$ and $X^{\uparrow}(n) = \{X_u^{\uparrow}(n); u \in [0,1]\}$ satisfying the monotonicity and limiting requirements (1.1) and (1.2) respectively. Note that X here is a continuous time Brownian bridge path, thus an infinite-dimensional random variable. However, the bounding processes will be piece-wise constant, thus inherently finite-dimensional. One will be able to realise complete sample paths of $X^{\downarrow}(n)$ or $X^{\uparrow}(n)$ on a computer without retreating to any sort of discretization or approximation errors (apart from those due to finite computing accuracy).

3.1. ϵ-Strong Algorithm

Given some initial intersection layer information $\mathcal{I}_{0,1}$, the algorithm will naturally set $X_u^{\uparrow}(0) = U_{0,1}^{\uparrow}$ and $X_u^{\downarrow}(0) = L_{0,1}^{\downarrow}$ for all instances $u \in [0,1]$. It will then iteratively *bisect* the acquired intersection layers, as described in Section 2.2, to obtain more information about the underlying sample path on finer time intervals. To ensure convergence of the discrepancy $X^{\uparrow}(n) - X^{\downarrow}(n)$ the algorithm will sometimes *refine* the information on some intersection layers, as described in Section 2.1, to reduce the uncertainty for the extrema. We give the pseudocode about the algorithm in Table 2.

Utilising the information the ϵ-strong algorithm returns, we define the dominating processes as follows:

$$\begin{aligned} X^{\uparrow}_u(n) &= \sum_{i=1}^{2^n} U^{\uparrow}_{(i-1)2^{-n},i\,2^{-n}} \cdot \mathrm{I}_{\,u\in((i-1)2^{-n},i\,2^{-n}\,]}\ , \\ X^{\downarrow}_u(n) &= \sum_{i=1}^{2^n} L^{\downarrow}_{(i-1)2^{-n},i\,2^{-n}} \cdot \mathrm{I}_{\,u\in((i-1)2^{-n},i\,2^{-n}\,]}\ . \end{aligned} \tag{3.1}$$

The square-root rate at Step 2.ii of the algorithm in Table 2 is to guarantee convergence of the dominating paths with minimal computing cost: it provides the correct distribution of effort between time-interval and extrema-interval bisections. To understand this, note that the range of a Brownian motion (or a Brownian bridge) on $[0,2^{-n}]$ scales as $\mathcal{O}(2^{-n/2})$; see for instance [18]. Thus, had we used the actual Brownian minima and maxima to define dominating processes for the Brownian path in the way of (3.1) the rate of convergence would have been $\mathcal{O}(2^{-n/2})$; we cannot exceed such a rate, but we can preserve it if our extrema are not further than $\mathcal{O}(2^{-n/2})$ from the actual ones. This intuitive statement will be made rigorous in the sequel, when an explicit result on the rate of convergence of the dominating processes in L_1-norm is given.

Fig.2 shows successive steps of the ϵ-strong algorithm as implemented on a computer. For each n, the red lines show the interval where the maxima and the minima are located: this information is available for all 2^n sub-intervals bisecting the initial time interval $[0,1]$. The black line corresponds to the linear interpolation of successively unveiled positions of the underlying Brownian path. The last graph (f) corresponds to $n=12$; in this case, we have zoomed on a particular subinterval of $[0,1]$ to be able to visualise the difference between the bounding paths and the underlying Brownian one.

3.2. Convergence Properties

Almost sure convergence of the dominating paths follows directly from the continuity of the Brownian path X. The analytical proof is given in the following proposition.

Proposition 3.1. *Consider the continuous-time processes $X^{\uparrow}(n)$, $X^{\downarrow}(n)$ defined in (3.1). Then, the convergence in supremum norm in (1.2) will hold in the limit $n\to\infty$.*

Proof. For a Brownian bridge X on $[0,1]$ we consider:

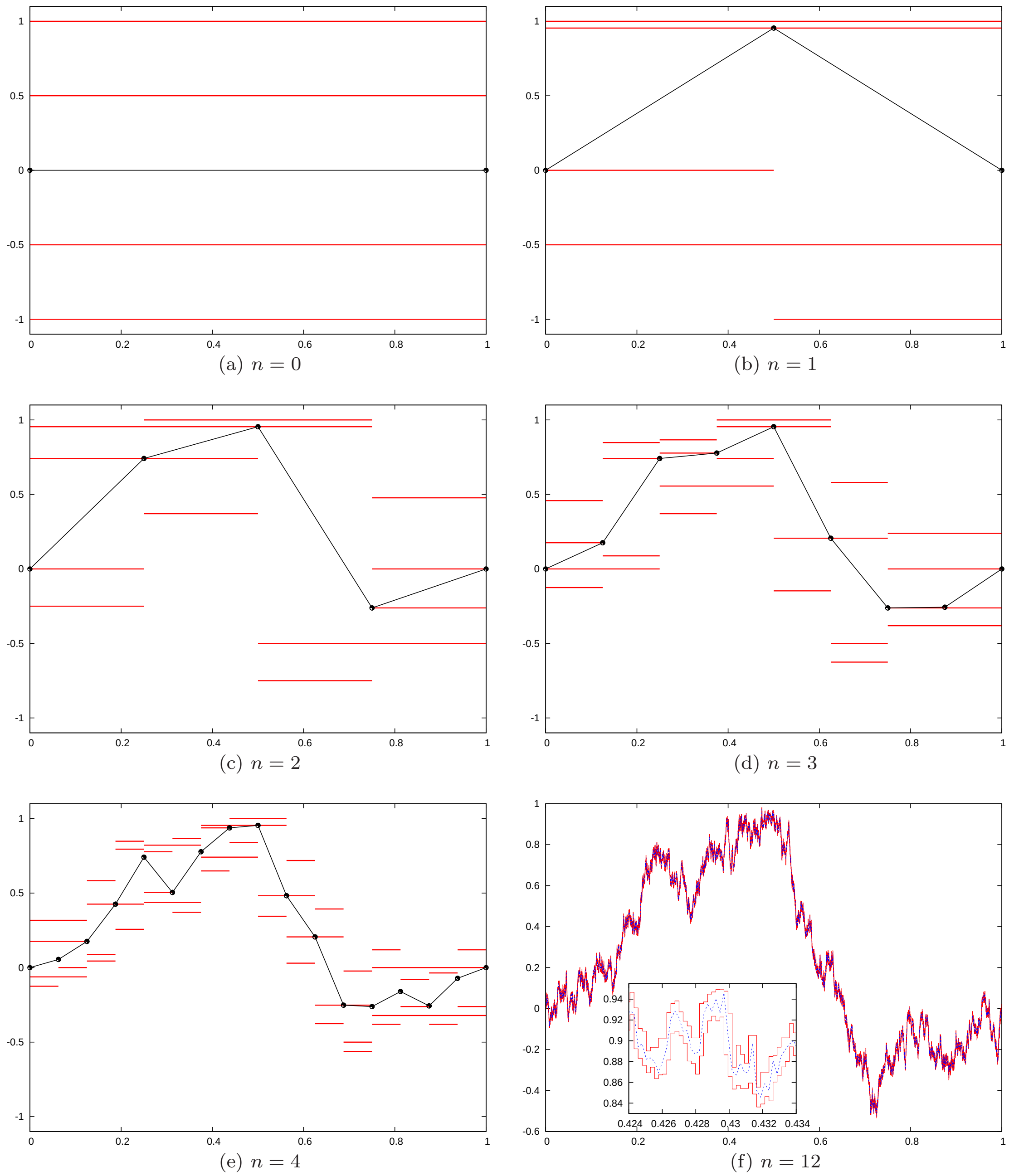
$$D_n := \sup_{1\le i\le 2^n} \left(M_{(i-1)2^{-n},i\,2^{-n}} - m_{(i-1)2^{-n},i\,2^{-n}}\right)\ .$$

Uniform continuity implies that, with probability 1:

$$\lim_{n\to\infty} D_n = 0\ .$$

Now, we have that:

$$\sup_{u\in[0,1]} |X^{\uparrow}_u(n) - X^{\downarrow}_u(n)| \le D_n + 2\cdot 2^{-n/2} \to 0\ ,$$



(a) $n = 0$

(b) $n = 1$

(c) $n = 2$

(d) $n = 3$

(e) $n = 4$

(f) $n = 12$

Figure 2. The ϵ-strong algorithm as applied on a personal computer. For each step n, the red lines show the allowed interval for the minima and the maxima: this information is separately available for all 2^n time sub-intervals partitioning $[0, 1]$. Note that the last graph corresponds to $n = 12$, with the subplot in its frame corresponding to a zooming on the position of the paths on the time interval $[0.424, 0.434]$.

where we have used the fact that Step 2.ii of the ϵ-strong algorithm guarantees that

$$U^{\uparrow}_{(i-1)2^{-n},i\,2^{-n}} \leq M_{(i-1)2^{-n},i\,2^{-n}} + 2^{-n/2} \ ,$$
$$L^{\downarrow}_{(i-1)2^{-n},i\,2^{-n}} \geq m_{(i-1)2^{-n},i\,2^{-n}} - 2^{-n/2} \ .$$

□

A more involved result can give the rate of convergence of the dominating processes and will be of practical significance for the efficiency of Monte-Carlo methods based on the ϵ-strong algorithm.

Proposition 3.2. *Consider the L_1-distance:*

$$|X^{\uparrow}(n) - X^{\downarrow}(n)|_1 = \int_0^1 |X^{\uparrow}_u(n) - X^{\downarrow}_u(n)|du \ .$$

Then:

$$2^{n/2} \times \mathrm{E}\left[\, |X^{\uparrow}(n) - X^{\downarrow}(n)|_1 \right] = \mathcal{O}(1) \ .$$

Proof. We proceed as follows:

$$\begin{aligned} |X^{\uparrow}(n) - X^{\downarrow}(n)|_1 &= \sum_{i=1}^{2^n} \big(U^{\uparrow}_{(i-1)2^{-n},i\,2^{-n}} - L^{\downarrow}_{(i-1)2^{-n},i\,2^{-n}} \big) \cdot 2^{-n} \\ &\leq \sum_{i=1}^{2^n} \Big(M_{(i-1)2^{-n},i\,2^{-n}} - m_{(i-1)2^{-n},i\,2^{-n}} + 2 \cdot 2^{-n/2} \Big) \cdot 2^{-n} \ , \end{aligned} \tag{3.2}$$

the inequality being a direct consequence of Step 2.ii of the ϵ-strong algorithm in Table 2. Consider now the path from $X_{(i-1)2^{-n}}$ to $X_{i\,2^{-n}}$. Let Z be a Brownian bridge from $Z_0 = 0$ to $Z_{2^{-n}} = 0$; we denote by M_z and m_z its maximum and minimum respectively. Conditionally on $X_{(i-1)2^{-n}}$ and $X_{i\,2^{-n}}$, a known property of the Brownian bridge implies (see e.g. [14]) that:

$$X_{t+(i-1)2^{-n}} = Z_t + (1 - \tfrac{t}{2^{-n}})\, X_{(i-1)2^{-n}} + \tfrac{t}{2^{-n}}\, X_{i\,2^{-n}}, \quad t \in [0, 2^{-n}] \ ,$$

in the sense that the processes on the two sides of the above equation have the same distribution. It is now clear that:

$$M_{(i-1)2^{-n},i\,2^{-n}} - m_{(i-1)2^{-n},i\,2^{-n}} \leq |X_{i\,2^{-n}} - X_{(i-1)2^{-n}}| + (M_z - m_z) \ .$$

So, taking expectations at (3.2), we get:

$$\mathrm{E}\left[\, |X^{\uparrow}(n) - X^{\downarrow}(n)|_1 \right] \leq \mathrm{E}\left[\, M_z - m_z \,\right] + 2 \cdot 2^{-n/2} + \sum_{i=1}^{2^n} \mathrm{E}\, |X_{i\,2^{-n}} - X_{(i-1)2^{-n}}|\, 2^{-n}$$

The finite-dimensional distributions of the initial Brownian bridge from X_0 to X_1 imply that:

$$X_{i\,2^{-n}} - X_{(i-1)2^{-n}} \sim N((X_1 - X_0)2^{-n}, 2^{-n}(1-2^{-n}))\ ,$$

which gives directly that:

$$\mathrm{E}\,|X_{i\,2^{-n}} - X_{(i-1)2^{-n}}| = \mathcal{O}(2^{-n/2})\ .$$

It remains to show that $\mathrm{E}\,[\,M_z - m_z\,] = \mathcal{O}(2^{-n/2})$ to complete the proof. Now, self-similarity of Brownian motion implies that:

$$Z_u = 2^{-n/2}\tilde{Z}_{u/2^{-n}}$$

where $\tilde{Z}$ is a Brownian bridge from $\tilde{Z}_0 = 0$ to $\tilde{Z}_1 = 0$. Let $\tilde{M}_z$, $\tilde{m}_z$ be the maximum and minimum of $\tilde{Z}$. Due to the self-similarity we have

$$M_z - m_z = 2^{-n/2}(\tilde{M}_z - \tilde{m}_z)\ .$$

Since $\tilde{M}_z - \tilde{m}_z$ in a random variable of finite expectation (see e.g. [14]), we obtain directly that $\mathrm{E}\,[\,M_z - m_z\,] = \mathcal{O}(2^{-n/2})$ which completes the proof.

□

4. The ζ-Function

We have yet to present the sampling methods employed when refining or bisecting an intersection layer during the execution of the ϵ-strong algorithm, thus constituting the building blocks of our algorithm. All probabilities involved in these methods can be expressed in terms of a hitting probability of the Brownian path. We denote by

$$\mathbb{W}^{(l,x,y)}$$

the probability law of a Brownian bridge from $X_0 = x$ to $X_l = y$. Let $\zeta(L,U;l,x,y)$, with $L < U$, be the probability that the Brownian bridge escapes the interval $[L,U]$. That is:

$$\zeta(L,U;l,x,y) = \mathbb{W}^{(l,x,y)}\,[\,m_{0,l} < L \text{ or } M_{0,l} > U\,]\ .$$

We also define:

$$\gamma(L,U;l,x,y) = 1 - \zeta(L,U;l,x,y)\ . \tag{4.1}$$

These probabilities can be calculated analytically in terms of an infinite series. The result is based on a partition of Brownian paths w.r.t. to a trace they leave on two bounding lines and can be attributed back to [11]; for more recent references see [1, 17, 9]. We define for $j \geq 1$,

$$\begin{aligned}\bar{\sigma}_j(x,y,\delta,\xi) &= \exp\left\{-\frac{2}{l}\,[\,\delta j + \xi - x\,]\,[\,\delta j + \xi - y\,]\right\},\\ \bar{\tau}_j(x,y,\delta) &= \exp\left\{-\frac{2j}{l}\,[\,\delta^2 j + \delta(x-y)\,]\right\}\ .\end{aligned} \tag{4.2}$$

Then, Theorem 3 of [17] yields

$$\zeta(L,U;l,x,y) = \begin{cases} \sum_{j=1}^{\infty}(\sigma_{_j} - \tau_{_j}) \ , & L < x, y < U \ , \\ 1 \ , & \text{otherwise} \ , \end{cases} \tag{4.3}$$

where

$$\begin{aligned} \sigma_j &= \bar{\sigma}_{_j}(x,y,U-L,L) + \bar{\sigma}_{_j}(-x,-y,U-L,-U) \ , \\ \tau_j &= \bar{\tau}_{_j}(x,y,U-L) + \bar{\tau}_{_j}(-x,-y,U-L) \ . \end{aligned} \tag{4.4}$$

The infinite series in (4.3) exhibits a monotonicity property which will be exploited by our simulation algorithms. We consider the sequence $\{S_n\}$, with $S_n = S_n(L,U;l,x,y)$, defined as:

$$S_{2n-1} = \sum_{j=1}^{n-1}(\sigma_j - \tau_j) + \sigma_n, \ \ S_{2n} = S_{2n-1} - \tau_n \ . \tag{4.5}$$

when $L < x, y < U$, otherwise $S_n \equiv 1$. Then:

$$0 < S_{2n} \leq S_{2n+2} \leq \zeta \leq S_{2n+1} \leq S_{2n-1} \tag{4.6}$$

for all $n \geq 1$; for a proof see [9] or [6].

4.1. ζ-Derived Events

We can combine ζ-probabilities to calculate other conditional probabilities arising in the context of the ϵ-strong algorithm. We begin with the following definition:

$$\beta(L^{\downarrow}, L^{\uparrow}, U^{\downarrow}, U^{\uparrow}; l, x, y) := \mathbb{W}^{(l,x,y)}\left[\, L^{\downarrow} < m_{0,l} < L^{\uparrow}, \, U^{\downarrow} < M_{0,l} < U^{\uparrow} \,\right] .$$

Now, we have the set equality:

$$\begin{aligned} \{L^{\downarrow} < m_{0,l} < L^{\uparrow}, \, U^{\downarrow} < M_{0,l} < U^{\uparrow}\} = \{L^{\downarrow} < m_{0,l}, \, M_{0,l} < U^{\uparrow}\} - \\ \{L^{\uparrow} < m_{0,l}, \, M_{0,l} < U^{\uparrow}\} \cup \{L^{\downarrow} < m_{0,l}, \, M_{0,l} < U^{\downarrow}\} \ . \end{aligned} \tag{4.7}$$

Thus, taking probabilities and recalling the definition of γ in (4.3), we find that:

$$\begin{aligned} \beta(L^{\downarrow}, L^{\uparrow}, U^{\downarrow}, U^{\uparrow}; l, x, y) &= \gamma(L^{\downarrow}, U^{\uparrow}; l, x, y) - \gamma(L^{\uparrow}, U^{\uparrow}; l, x, y) \\ &\quad - \gamma(L^{\downarrow}, U^{\downarrow}; l, x, y) + \gamma(L^{\uparrow}, U^{\downarrow}; l, x, y) \quad . \end{aligned} \tag{4.8}$$

Before the next event, we enrich the notation for the Brownian bridge measure. We define (for $0 < q < l$):

$$\mathbb{W}^{(l,x,y)}_{(q,w)}\left[\,\cdot\,\right] = \mathbb{W}^{(l,x,y)}\left[\,\cdot \mid X_q = w\,\right] .$$

We set $r = l - q$. Consider now the conditional probability:

$$\rho(L^{\downarrow}, L^{\uparrow}, U^{\downarrow}, U^{\uparrow}; q, r, x, w, y) = \mathbb{W}^{(l,x,y)}_{(q,w)}\left[\, L^{\downarrow} < m_{0,l} < L^{\uparrow}, \, U^{\downarrow} < M_{0,l} < U^{\uparrow} \,\right] .$$

Using again the set equality (4.7), and taking probabilities under $\mathbb{W}^{(l,x,y)}_{(q,w)}$, we obtain:

$$\rho(L^{\downarrow}, L^{\uparrow}, U^{\downarrow}, U^{\uparrow}; q, r, x, w, y) = \gamma_1\gamma_2 - \gamma_3\gamma_4 - \gamma_5\gamma_6 + \gamma_7\gamma_8 \tag{4.9}$$

where we have defined:

$$\gamma_1 = \gamma(L^{\downarrow}, U^{\uparrow}; q, x, w)\ ,\ \gamma_2 = \gamma(L^{\downarrow}, U^{\uparrow}; r, w, y)\ ,\ \gamma_3 = \gamma(L^{\uparrow}, U^{\uparrow}; q, x, w)\ ,$$
$$\gamma_4 = \gamma(L^{\uparrow}, U^{\uparrow}; r, w, y)\ ,\ \gamma_5 = \gamma(L^{\downarrow}, U^{\downarrow}; q, x, w)\ ,\ \gamma_6 = \gamma(L^{\downarrow}, U^{\downarrow}; r, w, y)\ ,$$
$$\gamma_7 = \gamma(L^{\uparrow}, U^{\downarrow}; q, x, w)\ ,\ \gamma_8 = \gamma(L^{\uparrow}, U^{\downarrow}; r, w, y)\ .$$

Note that the product terms arise due to the independency of the Brownian bridges on $[0, q]$ and $[q, l]$. We will be using these expressions for $\beta(\cdot;\cdot)$ and $\rho(\cdot;\cdot)$ in the sequel.

4.2. Simulation of ζ-Derived Events

We will need to be able to decide whether events of probability ζ have occurred or not. In a simulation context, this corresponds to determining the value of the binary variable $\mathrm{I}_{R<\zeta}$ for $R \sim \mathrm{Un}\,[\,0, 1\,]$. With (4.6) in mind, we define:

$$J = \inf\{n \geq 1 : n \text{ odd},\ S_n < R \text{ or } n \text{ even},\ S_n > R\}\ .$$

Due to the alternating monotonicity property (4.6) of S_n:

$$\mathrm{I}_{R<\zeta} = \mathrm{I}_{J \text{ is even}}\ .$$

Thus, we need a.s. finite number of J computations to evaluate $\mathrm{I}_{R<\zeta}$. Note that S_n converges to its limit exponentially fast, so J will be of small expectation; one can easily verify that all its moments are finite. Such an approach was also followed in [6].

In a more general context, we will also be required to decide if events of probability $\beta(\cdot;\cdot)$ or $\rho(\cdot;\cdot)$ have taken place or not; we will in fact be considering even more complex events related with the ζ-function. In the most encompassing scenario, when executing our sampling methods, we will be required to compare a given real number R with $Z(\zeta_1, \zeta_2, \ldots, \zeta_m)$ for some given function Z, with the different ζ_i's corresponding to different choices of the arguments l, x, y, L, U for $\zeta(\,\cdot\,;\cdot)$. Using the monotonicity property (4.6) we will be able to develop corresponding alternating sequences S^Z_n such that:

$$\begin{gathered} S^Z_{2n} \leq S^Z_{2n+2} \leq Z(\zeta_1, \zeta_2, \ldots, \zeta_m) \leq S^Z_{2n+1} \leq S^Z_{2n-1}\ ; \\ \lim_{n\to\infty} S^Z_n = Z(\zeta_1, \zeta_2, \ldots, \zeta_m)\ , \end{gathered} \tag{4.10}$$

and proceed as above. Analytically, we will determine the value of the comparison binary indicator $\mathrm{I}_{R<Z(\zeta_1,\zeta_2,\ldots,\zeta_m)}$ as follows:

Calculate S^Z_n until the first n such that either n is odd and $S^Z_n < R$ (whence return 0) or n is even and $S^Z_n > R$ (whence return 1).

5. Distributions and their Simulation

We will now describe analytically all simulation algorithms employed at the development of the ϵ-strong algorithm presented in Table 2. In particular, one has to develop sampling methods to carry out the refinement and bisection (see Section 2) of the intersection layer $\mathcal{I}_{s,t}$. To simplify the presentation, when conditioning on X_s, X_{t^*} or X_t we will make the correspondence:

$$x = X_s,\ w = X_{t^*},\ y = X_t\ ,$$
$$l = t - s,\ q = t^* - s,\ r = t - t^*\ .$$

5.1. Bisection of $\mathcal{I}_{s,t}$: Sampling the Middle Point X_{t^*}.

Bisection of $\mathcal{I}_{s,t} = \{X_s, X_t, \mathcal{L}_{s,t}, \mathcal{U}_{s,t}\}$, with $\mathcal{L}_{s,t} = [L^{\downarrow}, L^{\uparrow}]$, $\mathcal{U}_{s,t} = [U^{\downarrow}, U^{\uparrow}]$, begins by sampling a point of the Brownian bridge conditionally on the collected information about its minimum and maximum; this is Step 1 of Table 1. Such a conditional distribution is analytically tractable via Bayes' theorem.

Proposition 5.1. *The distribution $\mathbb{W}[\,X_{t^*} \mid \mathcal{I}_{s,t}\,]$, with $t^* \in [s,t]$, has probability density:*

$$f(w) \propto \rho(L^{\downarrow}, L^{\uparrow}, U^{\downarrow}, U^{\uparrow}; q, r, x, w, y) \times \pi(w)$$

where $\rho(\cdot;\cdot)$ is defined in (4.9) and

$$\pi(w) = \exp\{-\tfrac{1}{2}\big(w - (\tfrac{r}{l}\,x + \tfrac{q}{l}\,y)\big)^2/\big(\tfrac{qr}{l}\big)\}\ .$$

Proof. The function $\pi(w)$ corresponds to the prior (unnormalised) density for the middle point $X_t^* \mid X_s, X_t$ which is easily found to be normally distributed with mean and variance as implied by the expression for $\pi(w)$. So, following the definition of $\rho(\cdot;\cdot)$ in (4.9), the stated result is an application of Bayes' theorem. □

We will develop a method for sampling from $f(w)$. It is easy to construct an alternating series bounding $f(w)$. Let:

$$\zeta_i = 1 - \gamma_i\ , \quad 1 \le i \le 8,$$

for the eight γ-functions appearing at the definition of ρ in (4.9). Let $\{S_{i,n}\}_{n\ge 1}$ be the alternating series (4.6) for ζ_i, for each $1 \le i \le 8$; that is:

$$0 < S_{i,2n} \le S_{i,2n+2} \le \zeta_i \le S_{i,2n+1} \le S_{i,2n-1}\ , \tag{5.1}$$

with $\lim_{n\to\infty} S_{i,n} = \zeta_i$. Consider the sequence $\{S_n^Z\}$ defined as follows:

$$\begin{aligned} S_n^Z &= (1 - S_{1,n+1} - S_{2,n+1} + S_{1,n}S_{2,n}) - (1 - S_{3,n} - S_{4,n} + S_{3,n+1}S_{4,n+1}) \\ &\quad - (1 - S_{5,n} - S_{5,n} + S_{5,n+1}S_{6,n+1}) + (1 - S_{7,n+1} - S_{8,n+1} + S_{7,n}S_{8,n})\ . \end{aligned} \tag{5.2}$$

Due to (5.1), one can easily verify that $\{S_n^Z\}$ is an alternating sequence for $\rho(\cdot;\cdot)$, in the sense that:

$$S_{2n}^Z \leq S_{2n+2}^Z \leq \rho(L^{\downarrow}, L^{\uparrow}, U^{\downarrow}, U^{\uparrow}, q, r, x, w, y) \leq S_{2n+1}^Z \leq S_{2n-1}^Z \tag{5.3}$$

with $\lim_{n\to\infty} S_n^Z = \rho(L^{\downarrow}, L^{\uparrow}, U^{\downarrow}, U^{\uparrow}; q, r, x, w, y)$.

We exploit this structure to build a rejection sampler to draw from the density $f(w)$ in Proposition 5.1. We will use proposals from:

$$f_{2n+1}(w) = S_{2n+1}^Z(w) \times \pi(w) \ ,$$

where we have emphasized the dependence of S_{2n+1}^Z on the argument w. Note that the domain of both $f(w)$, $f_{2n+1}(w)$ is $[L^{\downarrow}, U^{\uparrow}]$. Now, we will illustrate that $S_{2n+1}^Z(w)$ has a concrete structure that we will exploit for our sampler. Consider the first of the four terms forming up S_{2n+1}^Z from (5.2):

$$1 - S_{1,2n+2} - S_{2,2n+2} + S_{1,2n+1}S_{2,2n+1} \ . \tag{5.4}$$

Following the analytical definition of the alternating sequences in equations (4.2), (4.4), (4.5), both $S_{1,n}$ and $S_{2,n}$, can be expressed as a sum of $2n$ terms each having the exponential structure $\pm\exp\{a + b\,w\}\,\mathrm{I}_{\,L^{\downarrow}<w<U^{\uparrow}}$ for appropriate constants a, b varying among the $2n$ terms. Thus, the quantity in (5.4) can be expressed as:

$$1 + \sum_{i=1}^{k_{1,n}} (-1)^{c_i} \exp\{a_i + b_i w\}\,\mathrm{I}_{\,L^{\downarrow}<w<U^{\uparrow}}$$

for $k_{1,n} = 4\{(2n+1)^2 + (2n+2)\}$, and constants a_i, b_i, c_i with $c_i \in \{0,1\}$. Working similarly for all four summands forming up S_{2n+1}^Z in (5.2) we get that the function $f_{2n+1}(w)$ can in fact be written as the weighted sum:

$$f_{2n+1}(w) = \sum_{i=1}^{k_n} (-1)^{c_i} \exp\{a_i + b_i w\}\,\mathrm{I}_{\,L_i<w<U_i} \times \pi(w) \tag{5.5}$$

for $k_n = 2(k_{1,n} + k_{2,n})$ with $k_{2,n} = 4\{(2n+2)^2 + (2n+1)\}$, and some explicit constants a_i, b_i, $c_i \in \{0,1\}$, L_i, U_i. Experimentation has showed that f_1 is already a very good envelope function for the rejection sampler, in which case $k_n \equiv k_0 = 64$; this is not accidental, and relates with the rapid exponential convergence of the alternating sequence in (4.5) to its limit. The cdf, say $F_1(w)$, corresponding to the unormalised density function $f_1(w)$ can be analytically identified since integrals for each of the summands in (5.5) can be expressed as differences of the cdf of the standard Gaussian distribution. Samples from $f_1(w)$ can then be generated using the inverse cdf method, i.e. by returning $F_1^{-1}(R)$ for $R \sim \mathrm{Un}[0,1]$. F_1^{-1} cannot be found analytically, but numerical methods can return $F_1^{-1}(R)$, up to maximum allowed computer accuracy, exponentially fast. We have used `MATHEMATICA` to automatically calculate all integrals giving the cdf, and then incorporated the calculation into a `C++` code.

Summarising, our rejection sampler will be as described below, where for simplicity we write $\rho(w) \equiv \rho(L^{\downarrow}, L^{\uparrow}, U^{\downarrow}, U^{\uparrow}; q, r, x, w, y)$:

Repeat until the first accepted draw:
Propose $w \sim f_1$ and accept with probability $f(w)/f_1(w) \equiv \rho(w)/S_1^Z(w)$.

Note here that the acceptance probability involves $\rho(w)$ which is made up of eight infinite series, see (4.9). We avoid approximations by using the alternating construction (5.3) and employ the methods of Section 4.2 to obtain the value of the decision indicator $\mathrm{I}_{R<\rho(w)/S_1^Z(w)}$ for some $R \sim \mathrm{Un}\,[\,0,1\,]$.

As shown in Step 1 of Table 1, once X_{t^*} is obtained, we adjust the allowed range for the extrema of the bridge on $[s,t]$ by simply setting $U^{\downarrow} = U^{\downarrow} \vee X_{t^*}$, $L^{\uparrow} = L^{\uparrow} \wedge X_{t^*}$.

5.2. Bisection of $\mathcal{I}_{s,t}$: Updating the Layers Given X_{t^*}.

At the second step of the bisection procedure, see Table 1, we obtain separate information for the extrema of the two newly formed bridges given the middle point X_{t^*}: the one bridge being from X_s to X_{t^*}, the other from X_{t^*} to X_t. In particular, the algorithm will decide over the range of the four newly formed layers, $\mathcal{L}_{s,t^*}$, $\mathcal{U}_{s,t^*}$, $\mathcal{L}_{t^*,t}$, $\mathcal{U}_{t^*,t}$ in the following manner: for the case of $\mathcal{L}_{s,t^*}$ for instance a decision will be made over whether m_{s,t^*} lies in $[L^{\downarrow}, L^{\uparrow}]$ (which is the allowed range for the minimum of the original bridge on $[s,t]$) or in $[L^{\uparrow}, X_s \wedge X_{t^*}]$. The apparent analogues apply in the case of the three other layers.

One might initially think that there are in total 2^4 different scenaria for the four layers. But one has to remember that the update has to respect the information in $\mathcal{I}_{s,t}$, so that at least one of the two minima (resp. maxima) on $[s,t^*]$ and $[t^*,t]$ *must* lie in $[L^{\downarrow}, L^{\uparrow}]$ (resp. $[U^{\downarrow}, U^{\uparrow}]$). In particular, there are in fact nine different possible scenaria, which are the ones shown in Table 3 (labelled as events $\{E = i\}$, for $1 \le i \le 9$): a value of 1 in Table 3 means that the corresponding minimum or maximum will still be found within the allowed range for the original bridge on $[s,t]$, whereas a value of 0 means that the second option occurs and the extremum will be shifted inwards. For instance, a value of 0 for the indicator variable concerning m_{s,t^*}, M_{s,t^*}, $m_{t^*,t}$ or $M_{t^*,t}$ implies that $m_{s,t^*} \in [L^{\uparrow}, X_s \wedge X_{t^*}]$, $M_{s,t^*} \in [X_s \vee X_{t^*}, U^{\downarrow}]$, $m_{t^*,t} \in [L^{\uparrow}, X_{t^*} \wedge X_t]$ or $M_{t^*,t} \in [X_{t^*} \vee X_t, U^{\downarrow}]$ respectively.

The probability for each of the events in Table 3 can be derived via functions $\beta(\cdot;\cdot)$ and $\rho(\cdot;\cdot)$ defined in (4.8) and (4.9) respectively. Recall that we are conditioning upon $\mathcal{I}_{s,t}$ and X_{t^*}, so we work as follows:

$$\begin{aligned}
\mathrm{P}\,[\,E = i\,|\,\mathcal{I}_{s,t}, X_{t^*}\,] &= \mathrm{P}\left[\,E = i \mid m \in [L^{\downarrow}, L^{\uparrow}],\, M \in [U^{\downarrow}, U^{\uparrow}],\, X_s,\, X_{t^*},\, X_t\,\right] \\
&= \frac{\mathrm{P}\left[\,E = i,\, m \in [L^{\downarrow}, L^{\uparrow}],\, M \in [U^{\downarrow}, U^{\uparrow}] \mid X_s,\, X_{t^*},\, X_t\,\right]}{\mathrm{P}\left[\,m \in [L^{\downarrow}, L^{\uparrow}],\, M \in [U^{\downarrow}, U^{\uparrow}] \mid X_s,\, X_{t^*},\, X_t\,\right]} \\
&= \frac{\mathrm{P}\left[\,E = i \mid X_s,\, X_{t^*},\, X_t\,\right]}{\rho(L^{\downarrow}, L^{\uparrow}, U^{\downarrow}, U^{\uparrow}; q, r, x, w, y)}\ .
\end{aligned}$$

Now, conditionally on $\{X_s,\, X_{t^*},\, X_t\}$ the law of the path factorises into two independent Brownian bridges. Thus, recalling also the definition of $\beta(\cdot;\cdot)$ in (4.8), the probability

$\mathrm{P}\,[\,E = i\,|\,X_s,\ X_{t^*},\ X_t\,]$ in the numerator above can be written as a product of two $\beta(\cdot;\cdot)$ functions. The analytical calculation of the numerator, or equivalently of the product $\mathrm{P}\,[\,E = i\,|\,\mathcal{I}_{s,t}, X_{t^*}\,] \times \rho(L^{\downarrow}, L^{\uparrow}, U^{\downarrow}, U^{\uparrow}; q, r, x, w, y)$, is given in Table 4 where, to simplify the presentation, we have set:

$$w_x = x \wedge w,\ \ w^x = x \vee w,\ \ w_y = w \wedge y,\ \ w^y = w \vee y\ .$$

The method to simulate the discrete random variable E could follow the alternating series approach of Section 4.2. Analytically, consider the cumulative probability values $p_i = \mathrm{P}\,[\,E \leq i\,|\,\mathcal{I}_{s,t}, X_{t^*}]$. A simple inverse cdf sampling method requires finding the index $\inf\{i \geq 1 : R < p_i\}$ for a $R \sim \mathrm{Un}\,[0,1]$. Note now that the p_i's can be bounded above and below by monotone converging sequences as in (4.10), thus each comparison $\{R < p_i\}$ can be carried out via the alternating series approach of Section 4.2 without any need for approximations.

Event	Left bridge		Right bridge	
$E = i$	$\mathrm{I}_{\,m_{s,t^*} \in [L^{\downarrow}, L^{\uparrow}]}$	$\mathrm{I}_{\,M_{s,t^*} \in [U^{\downarrow}, U^{\uparrow}]}$	$\mathrm{I}_{\,m_{t^*,t} \in [L^{\downarrow}, L^{\uparrow}]}$	$\mathrm{I}_{\,M_{t,t^*} \in [U^{\downarrow}, U^{\uparrow}]}$
$i = 1$	1	1	1	1
$i = 2$	1	1	0	1
$i = 3$	1	1	1	0
$i = 4$	1	1	0	0
$i = 5$	0	1	1	1
$i = 6$	0	1	1	0
$i = 7$	1	0	1	1
$i = 8$	1	0	0	1
$i = 9$	0	0	1	1

Table 3. The nine possible scenaria for the extrema of the two Brownian bridges (from X_s to X_t^* and from X_{t^*} to X_t).

i	$\mathrm{P}\,[\,E = i\,\|\,\mathcal{I}_{s,t}, X_{t^*}\,] \times \rho(L^{\downarrow}, L^{\uparrow}, U^{\downarrow}, U^{\uparrow}; q, r, x, w, y)$
1	$\beta(L^{\downarrow}, L^{\uparrow}, U^{\downarrow}, U^{\uparrow}; q, x, y) \times \beta(L^{\downarrow}, L^{\uparrow}, U^{\downarrow}, U^{\uparrow}; r, w, y)$
2	$\beta(L^{\downarrow}, L^{\uparrow}, U^{\downarrow}, U^{\uparrow}; q, x, w) \times \beta(L^{\uparrow}, w_y, U^{\downarrow}, U^{\uparrow}; r, w, y)$
3	$\beta(L^{\downarrow}, L^{\uparrow}, U^{\downarrow}, U^{\uparrow}; q, x, w) \times \beta(L^{\downarrow}, L^{\uparrow}, w^y, U^{\downarrow}; r, w, y)$
4	$\beta(L^{\downarrow}, L^{\uparrow}, U^{\downarrow}, U^{\uparrow}; q, x, w) \times \beta(L^{\uparrow}, w_y, w^y, U^{\downarrow}; r, w, y)$
5	$\beta(L^{\uparrow}, w_x, U^{\downarrow}, U^{\uparrow}; q, x, w) \times \beta(L^{\downarrow}, L^{\uparrow}, U^{\downarrow}, U^{\uparrow}; r, w, y)$
6	$\beta(L^{\downarrow}, L^{\uparrow}, U^{\downarrow}, U^{\uparrow}; q, x, w) \times \beta(L^{\downarrow}, L^{\uparrow}, U^{\downarrow}, U^{\uparrow}; r, w, y)$
7	$\beta(L^{\uparrow}, w_x, U^{\downarrow}, U^{\uparrow}; q, x, w) \times \beta(L^{\downarrow}, L^{\uparrow}, w^y, U^{\downarrow}; r, w, y)$
8	$\beta(L^{\downarrow}, L^{\uparrow}, w^x, U^{\downarrow}; q, x, w) \times \beta(L^{\uparrow}, w_y, U^{\downarrow}, U^{\uparrow}; r, w, y)$
9	$\beta(L^{\uparrow}, w_x, w^x, U^{\downarrow}; q, x, w) \times \beta(L^{\downarrow}, L^{\uparrow}, U^{\downarrow}, U^{\uparrow}; r, w, y)$

Table 4. The conditional probabilities for each of the events in Table 3.

5.3. Remaining Sampling Procedures

A sampling algorithm is required for the refinement of the uncertainty over the extrema of a Brownian bridge. As described in Section 2.1, given the current intersection layer information $\mathcal{I}_{s,t}$ and in particular the fact that $M_{s,t} \in [L^{\downarrow}, L^{\uparrow}]$, the algorithm will need to decide whether the maximum $M_{s,t}$ lies in $[U^{\downarrow}, U^{*}]$ or in $[U^{*}, U^{\uparrow}]$, for $U^{*} = (U^{\downarrow} + U^{\uparrow})/2$. Recalling the definition of $\beta(\cdot;\cdot)$ from (4.8), it is easy to check that the ratio:

$$\frac{\beta(L^{\downarrow}, L^{\uparrow}, U^{*}, U^{\uparrow}; l, x, y)}{\beta(L^{\downarrow}, L^{\uparrow}, U^{\downarrow}, U^{\uparrow}; l, x, y)}$$

provides precisely the probability of the event $\{M_{s,t} \in [U^{*}, U^{\uparrow}] \,|\, \mathcal{I}_{s,t}\}$. Thus, we can again use the alternating sequence construction of Section 4.2 to simulate, without approximation, the binary variable $\mathrm{I}_{\,M_{s,t} \in [U^{*}, U^{\uparrow}]}$. The same approach can be followed for refining the allowed range for the minimum $m_{s,t}$.

We should also give some details over the initialization of the layers $\mathcal{U}_{0,1}$ and $\mathcal{L}_{0,1}$ at the first step of the ϵ-strong algorithm in Table 2 given X_0 and X_1. (Note that sometimes, as in the example applications that we consider in the following section, this initialization steps might not even be necessary, as the problem at hand provides a natural definition of $\mathcal{U}_{0,1}$ and $\mathcal{L}_{0,1}$.) One way to proceed is by specifying increasing sequences $\{a_i\}_{i \geq 0}$, $\{b_i\}_{i \geq 0}$, with $a_0 = b_0 = 0$, growing to ∞ and a bivariate index I such that:

$$\{I = (i,j)\} = \{\bar{x} - a_i < m \leq \bar{x} - a_{i-1},\, \bar{y} + b_{j-1} < M \leq \bar{y} + b_j\}\ ,$$

where $\bar{x} = x \wedge y$, $\bar{y} = x \vee y$. We can easily identify the probability distribution of I under the Brownian bridge dynamics since:

$$\mathbb{W}^{(1,x,y)}\,[\, I = (i,j)\,] = \beta(\bar{x} - a_i, \bar{x} - a_{i-1}, \bar{y} + b_{j-1}, \bar{y} + b_j; 1, x, y)\ .$$

Thus, we can work as in the case of the simulation of the discrete variable E in Section 5.2: assuming $\tilde{I} = 1, 2, \ldots$ is some chosen ordering of the states of I, an inverse cdf method would required finding $\inf\{i \geq 1 : R < \mathrm{P}\,[\, I \leq i\,]\}$ for $R \sim \mathrm{Un}[0,1]$, and approximations at the comparison between R and $\mathrm{P}\,[\, I \leq i\,]$ can be avoided via the alternating series approach. In practice, one could select some big enough values for the first couple of elements of the sequences $\{a_i\}$ and $\{b_i\}$ so that almost all probability mass is concentrated on $\{I = (i,j)\}$ for $i, j \leq 2$, and not a lot of computational resources are spent on this step.

6. Application: Unbiased Estimation of Path Expectations

The information provided by the ϵ-strong algorithm can be exploited to deliver *unbiased* estimators for path expectations arising in applications, avoiding discretization errors characterising standard approaches. We emphasize that we mean to sketch here only a potential direction for application of the algorithm. Analytically, consider a non-negative path functional $F : C([0,1], \mathbb{R}) \mapsto \mathbb{R}^{+}$ and the expectation: $\mathrm{E}\,[\, F(X)\,]$, X being a Brownian motion on $[0,1]$. One can easily check, by integrating out E, that:

$$\mathrm{I}_{\,F(X) > E} \cdot e^{E}\ , \quad E \sim \mathrm{Exp}(1)\ , \tag{6.1}$$

with E being independent of X, is an unbiased estimator of $\mathrm{E}\,[\,F(X)\,]$. The ϵ-strong algorithm could be utilised here to unbiasedly obtain the value of the binary variable $\mathrm{I}_{\,F(X)>E}$ in finite computations. We can easily find the second moment of the unbiased estimator in (6.1):

$$\mathrm{E}\,[\,\mathrm{I}_{\,F(X)>E}\cdot e^{2E}\,] = \mathrm{E}\,[\,e^{F(X)}\,] - 1\ . \tag{6.2}$$

We describe for a moment in more detail the identification of $\mathrm{I}_{F(X)>U}$ via the ϵ-strong algorithm. Utilising the lower and upper convergent processes $X^{\downarrow}(n)$, $X^{\uparrow}(n)$ in (3.1) one could in many cases analytically identify quantities $F_n^{\downarrow}$, $F_n^{\uparrow}$ (realisable with finite computations) such that:

$$\begin{gathered} F_n^{\downarrow} \le F_{n+1}^{\downarrow} \le F(X) \le F_{n+1}^{\uparrow} \le F_n^{\uparrow}\,; \\ F_n^{\uparrow} - F_n^{\downarrow} \to 0\ . \end{gathered}$$

Given enough iterations, there will be agreement; for the a.s. finite random instance:

$$\kappa = \inf\{n\ge 0:\ \mathrm{I}_{\,F_n^{\downarrow}>E} = \mathrm{I}_{\,F_n^{\uparrow}>E}\,\} \tag{6.3}$$

we will have

$$\mathrm{I}_{\,F(X)>E} = \mathrm{I}_{\,F_\kappa^{\downarrow}>E}\ . \tag{6.4}$$

Thus, combining (6.1) with (6.4), we have developed an unbiased estimator of a path expectation, involving finite computations. Certainly, the numerical efficiency of such an estimation will rely heavily on the stochastic properties of κ and the cost of generating $F_n^{\downarrow}$, $F_n^{\uparrow}$, and of course the variance of the estimator.

The particular derivation of the above unbiased estimator of the path expectation is by no means restrictive; one can generate unbiased estimators using distributions other than the exponential. Consider the following scenario. We can generate some preliminary bounds $F_{n_0}^{\downarrow}$, $F_{n_0}^{\uparrow}$ up to some fixed or random (depending on X) instance n_0. Now, one can easily check (by considering the conditional expectation w.r.t. $R|X$) that:

$$\mathrm{I}_{F(X)>R}\,F_{n_0}^{\uparrow} + \mathrm{I}_{F(X)<R}F_{n_0}^{\downarrow};\quad R\sim \mathrm{Un}[F_{n_0}^{\downarrow}, F_{n_0}^{\uparrow}]\ , \tag{6.5}$$

is also an unbiased estimator of $\mathrm{E}\,[\,F(X)\,]$. We have empirically found the estimator (6.5) to be much more robust than (6.1) in the numerical applications we present in the sequel. This is not accidental: for instance, considering a random n_0 such that $F_{n_0}^{\uparrow} - F_{n_0}^{\downarrow} < C$, for a constant $C>0$, we get that the second moment of the estimator (6.5) will be:

$$\begin{aligned} &\mathrm{E}\,[\,\mathrm{I}_{F(X)>R}\,(F_{n_0}^{\uparrow})^2 + \mathrm{I}_{F(X)<R}(F_{n_0}^{\downarrow})^2\,] = \\ &\qquad = \mathrm{E}\,[\,F(X)(F_{n_0}^{\uparrow}+F_{n_0}^{\downarrow})\,] - \mathrm{E}\,[\,F_{n_0}^{\uparrow}F_{n_0}^{\downarrow}\,] \le \mathrm{E}\,[\,F^2(X)\,] + C\,\mathrm{E}\,[\,F(X)\,] \end{aligned}$$

which has now a quadratic structure - compare this with (6.2). In general, increasing n_0 adds to the computational cost per sample, but decreases the variance. We have empirically found that moderate values of n_0 deliver significantly better estimates than (6.1) and will be using such an approach for our numerical examples in the sequel.

6.1. Numerical Illustrations

We will apply the ϵ-strong algorithm to unbiasedly estimate some option prices arising in finance. In particular option prices are expressed as expectations:

$$\mathrm{E}\,[\,F(S)\,]$$

of a functional $F(\cdot)$ of the path process $S = \{S_t\}$ modelling the underlying asset. We will consider some double-barrier options corresponding the expectations of the functionals:

$$F_a(S) = e^{-rT}(\,\sup S_t - K_S\,)^+\;\mathrm{I}_{\,L_S<\inf S_t<\sup S_t<U_S}\;; \tag{6.6}$$

$$F_b(S) = e^{-rT}\big(\,\tfrac{1}{T}\int_0^T S_t\,dt - K_S\big)^+\;\mathrm{I}_{\,L_S<\inf S_t<\sup S_t<U_S}\;; \tag{6.7}$$

$$F_c(S) = e^{-rT}(\,\sup S_t - K_S\,)^+\;\mathrm{I}_{\,L_S<\inf S_t\,e^{-rt}<\sup S_t\,e^{-rt}<U_S}\;, \tag{6.8}$$

(where for $x \in \mathbb{R}$, $x^+ := x \vee 0$) for underlying asset $S = \{S_t; t \in [0,T]\}$ modelled via a geometric Brownian motion (we consider a Black and Scholes framework) determined as:

$$\log S_t = \log S_0 + (r - \tfrac{\sigma^2}{2})\,t + \sigma\,W_t \tag{6.9}$$

for constants r (interest rate), $\sigma > 0$ (volatility) and a Brownian motion $\{W_t\}$. Also, T above is the maturity time, K_S the strike price and L_S, U_S the lower and upper barriers respectively; suprema and infima are considered over the time period $[0,T]$. Note that $\mathrm{E}[\,F_b(S)\,]$ corresponds to the price of the Asian option, see e.g. [21]. The process S_t is an 1-1 transformation of a Brownian motion with drift. In particular, we can rewrite the functionals (6.6)-(6.8) as follows:

$$F_a(X) = e^{-rT}(\,e^{\sigma\sup X_t} - K_S\,)^+\;\mathrm{I}_{\,L<\inf X_t<\sup X_t<U}\;; \tag{6.10}$$

$$F_b(X) = e^{-rT}\big(\,\tfrac{1}{T}\int_0^T e^{\sigma X_t}\,dt - K_S\big)^+\;\mathrm{I}_{\,L<\inf X_t<\sup X_t<U}\;; \tag{6.11}$$

$$F_c(X) = e^{-rT}(\,e^{\sigma\sup(\frac{r}{\sigma}\,t+X_t)} - K_S\,)^+\;\mathrm{I}_{\,L<\inf X_t<\sup X_t<U}\;, \tag{6.12}$$

for $L = \log(L_S)/\sigma$, $U = \log(U_S)/\sigma$, and:

$$\begin{aligned}\text{Case } F_a, F_b: &\quad X_t = \log(S_0)/\sigma + (\tfrac{r}{\sigma} - \tfrac{\sigma}{2})t + W_t\;;\\ \text{Case } F_c: &\quad X_t = \log(S_0)/\sigma - \tfrac{\sigma}{2}\,t + W_t\;.\end{aligned}$$

Conditionally on its ending point, the dynamics of the drifted Brownian motion do not depend on the value of the drift and coincide with those of a simple Brownian bridge; this is a simple by-product of the Girsanov theorem, see e.g. [15]. Thus, the ϵ-strong algorithm can now deliver convergent, lower and upper dominating processes for X.

The choices of functionals in (6.6)-(6.8) is not accidental: some generic characteristics of the structure of each functional (relevant also for other applications) will effect the set-up of the ϵ-strong algorithm and its efficiency; we will say more on this in the sequel.

For all three examples, our general methodology is as follows: we begin by sampling X_T and, then, the indicator variable $\mathrm{I}_{\,L<\inf X_t<\sup X_t<U}$; if the latter is 0 the sample for our unbiased estimator is simply 0, otherwise we proceed with applying the methods of the ϵ-strong algorithm by initializing the first intersection layer as $\mathcal{I}_{0,T} = \{X_0, X_T, \mathcal{U}_{0,T}, \mathcal{L}_{0,T}\}$ for intervals $\mathcal{U}_{0,T} = [X_0 \vee X_T, U]$ and $\mathcal{L}_{0,T} = [L, X_0 \wedge X_T]$. In some cases we might not need all of the machinery of the ϵ-strong algorithm to construct the sequences $F_n^{\downarrow}$, $F_n^{\uparrow}$ enveloping $F(X)$, with direct implications on the efficiency of the algorithm, as we explain analytically below.

F_a-example: Only Refinement

Here, we need information only on the marginal variable $\sup X_t$ (and not the whole of the continuous path on $[0,T]$) to develop an alternating series for $F_a(X)$. Thus, it suffices to apply a reduced version of the complete ϵ-strong algorithm in Table 2 where we only repeatedly refine the initial intersection layer $\mathcal{I}_{0,T}$ (in particular, we only refine the layer for the maximum) as described in Section 2.1 (and never bisect it) to construct $F_n^{\downarrow}$, $F_n^{\uparrow}$. In particular, having defined:

$$\phi(x) = e^{-rT}(\, e^{\sigma x} - K_S\,)^+ \ .$$

knowing that after n refinements the allowed range for $\sup X_t$ is $[U_n^{\downarrow}, U_n^{\uparrow}]$ (with initial position $[U_0^{\downarrow}, U_0^{\uparrow}] = [\, X_0 \vee X_T, U])$ we set $F_n^{\downarrow} = \phi(U_n^{\downarrow})$, $F_n^{\uparrow} = \phi(U_n^{\uparrow})$.

F_b-example: Refinement And Bisection

The complete machinery of the ϵ-strong algorithm in Table 2 is required here as we need to bound a path integral. Recall that the n-th step of the algorithm provides the piecewise constant paths $X_t^{\downarrow}(n)$, $X_t^{\uparrow}(n)$ enveloping X defined in (3.1). We now set $F_n^{\downarrow} = F_b(X^{\downarrow}(n))$, $F_n^{\uparrow} = F_b(X^{\uparrow}(n))$.

F_c-example: Selective Refinement And Bisection

We will now only need to bisect a *selection* of intersection layers as we will be allowed to delete intersection layers that cannot definitely contain $\sup(\frac{r}{\sigma}\,t + X_t)$ during the execution of the ϵ-strong algorithm. In particular, assuming the current collection (after $n-1$ steps) of stored intersection layers $\mathcal{I}_{s_i,t_i}$, with $s_i < t_i \leq s_{i+1}$, containing information about the path X, and determining the allowed range for $\sup(\frac{r}{\sigma}\,t + X_t)$:

$$[\, U_{n-1}^{\downarrow}, U_{n-1}^{\uparrow}\,] = [\, \sup_i\{U_{s_i,t_i}^{\downarrow} + \tfrac{r}{\sigma}\, s_i\},\, \sup_i\{U_{s_i,t_i}^{\uparrow} + \tfrac{r}{\sigma}\, t_i\}\,] \tag{6.13}$$

we proceed to the n-th step where: (i) we bisect and refine all stored intersection layers $\mathcal{I}_{s_i,t_i}$, (ii) calculate the running bounds $[\, U_n^{\downarrow}, U_n^{\uparrow}\,]$ by taking the suprema as in (6.13) but now over all newly obtained intersection layers, (iii) *delete* the obtained intersection layers $\mathcal{I}_{s,t}$ for which $U_{s,t}^{\uparrow} < U_n^{\downarrow}$ (as they cannot offer extra information on the whereabouts of $\sup(\frac{r}{\sigma}\,t + X_t)$ given that we already know that the latter is within $[\, U_n^{\downarrow}, U_n^{\uparrow}\,]$) and store *only* the remaining ones for the next iteration. At each step we set $F_n^{\downarrow} = \phi(U_n^{\downarrow})$, $F_n^{\uparrow} = \phi(U_n^{\uparrow})$ with ϕ as defined above.

Numerics

We have run the ϵ-strong algorithm for the above scenaria. To give an idea of its computing cost, we compare its execution times with those of the standard (Euler) approximation method that replaces the continuous-time path $\{S_t\,;\, t \in [0,T]\}$ with its discretised approximation $\{S_{t_i}\}_{i=0}^{l}$, for $t_i = t_{i-1} + \delta$, with step-size $\delta = T/l$; then, continuous-time maxima and integrals appearing in the functionals (6.6)-(6.8) are replaced with their obvious approximations based on the discrete-time vector $\{S_{t_i}\}$. We run our simulations under the parameter selections:

$$r = 0.05\ ,\ \sigma = 0.2\ ,\ S_0 = 1\ ,\ K = 1\ ,\ T = 1\ ,\ U = 1.25\ ,\ L = 0.75\ .$$

Tables 5-7 show results from the simulation study. For each different algorithm we show its execution time (all algorithms were coded in `C++`) and a 95% confidence interval for the mean of the realised estimators to give an idea about the variance of the estimates and their bias (for the case of the Euler approximation, as the ϵ-strong algorithm is unbiased). The results in Tables 5, 7 are obtained via 100,000 independent realizations of the estimators, whereas those in Table 6 via 10,000 independent realizations.

Euler Approximation		
δ	time (secs)	95% Conf. Int.
1/10	0.4	$[638, 647] \times 10^{-4}$
1/20	0.8	$[657, 667] \times 10^{-4}$
1/40	1.5	$[669, 679] \times 10^{-4}$
1/80	2.9	$[674, 683] \times 10^{-4}$
1/160	5.8	$[680, 689] \times 10^{-4}$

ϵ-strong		
n_0	time (secs)	95% Conf. Int.
2	1.1	$[683, 693] \times 10^{-4}$

Table 5. Simulation results from the application of the Euler approximation and the ϵ-strong algorithm for the estimation of the option price in $\mathrm{E}\,[\,F_a(S)\,]$ in (6.6). The results in the table correspond to a sample of 100,000 estimates. δ is the discretisation increment of the Euler method, and n_0 is the number of preliminary steps for the ϵ-strong algorithm before the simulation of the uniform random variable (see (6.5)).

Euler Approximation		
δ	time (secs)	95% Conf. Int.
10^{-1}	0.1	$[157, 169] \times 10^{-4}$
10^{-2}	0.4	$[120, 130] \times 10^{-4}$
10^{-3}	3.5	$[106, 116] \times 10^{-4}$
10^{-4}	34.3	$[107, 116] \times 10^{-4}$
10^{-5}	344.8	$[102, 112] \times 10^{-4}$

ϵ-strong		
n_0	time (secs)	95% Conf. Int.
2	115.9	$[81, 128] \times 10^{-4}$

Table 6. Similar results as for Table 5, but now for the case of the Asian option $\mathrm{E}\,[\,F_b(S)\,]$ in (6.7) - with the difference that here the results correspond to a sample of 10,000 estimates.

Looking at the three tables, we can make some comments; we focus more here on giving a simple picture to the reader than being mathematically precise. The cost *per sample* of the ϵ-strong algorithm corresponds to that of the Euler approximation with $\delta \approx 1/40$, $\delta \approx 4^{-1}{\cdot}10^{-4}$ and $\delta \approx 10^{-3}$ for the cases of $\mathrm{E}\,[\,F_a(S)\,]$, $\mathrm{E}\,[\,F_b(S)\,]$ and $\mathrm{E}\,[\,F_c(S)\,]$ respectively. Taking also the standard deviation under consideration (but *not* the bias)

Euler Approximation		
δ	time (secs)	95% Conf. Int.
1/10	0.5	$[797, 807] \times 10^{-4}$
1/20	0.9	$[822, 832] \times 10^{-4}$
1/40	1.7	$[833, 844] \times 10^{-4}$
1/80	3.3	$[835, 846] \times 10^{-4}$
1/160	6.6	$[846, 858] \times 10^{-4}$

ϵ-strong		
n_0	time (secs)	95% Conf. Int.
2	178.0	$[842, 854] \times 10^{-4}$

Table 7. Similar results as for Table 5, but now for the case of $\mathrm{E}\,[\,F_c(S)\,]$ in (6.7).

from the column with the confidence intervals, for the case of $\mathrm{E}\,[\,F_b(S)\,]$ we would need about 25 times more samples than then Euler approximation to attain the same range for the confidence interval; thus, ignoring the bias for the Euler approach, one could say that the overall cost of the ϵ-strong algorithm for the case of $\mathrm{E}\,[\,F_b(S)\,]$ corresponds to that of the Euler method with step-size $\delta' \approx (4 \cdot 25)^{-1}10^{-4} = 10^{-6}$.

However, a general remark here is that the ϵ-strong algorithm returns unbiased estimators of the relevant path expectations, and for the applications we have considered above it can provide accurate, unbiased estimates in reasonable amounts of time. Even when ignoring the bias of the Euler approach, for the cases of $\mathrm{E}\,[\,F_a(S)\,]$ and $\mathrm{E}\,[\,F_c(S)\,]$ the cost of the ϵ-strong algorithm already seems to be on a par with that of the Euler method for relatively non-conservative choices of discretisation step δ. (We should also stress that there is definitely great space for improving the efficiency of the used computing code for the ϵ-strong algorithm.)

6.2. Remark on Number of Bisections for ϵ-Strong Algorithm.

We make a comment here on the number of required iterations before the value of the binary variable $\mathrm{I}_{F(X)>R}$ in (6.5) is decided. Proposition 3.2 will be of relevance in this context. Recall that κ in (6.3) denotes the number of steps to decide about $\mathrm{I}_{F(X)>R}$. The cost of κ iterations of the ϵ-strong algorithm (when its full machinery is required) is proportional to $\mathcal{K} = 2^{\kappa}$. In the context of (6.5), we find:

$$\mathrm{P}\,[\,\mathcal{K} > 2^n\,] = \mathrm{E}\,[\,\mathrm{P}\,[\,\kappa > n \mid X\,]\,] = \mathrm{E}\,\Big[\,\frac{F_n^{\uparrow} - F_n^{\downarrow}}{F_{n_0}^{\uparrow} - F_{n_0}^{\downarrow}}\,\Big]\ .$$

Proposition 3.2 states that $|X^{\uparrow}(n) - X^{\downarrow}(n)|_{L_1} = \mathcal{O}(2^{-n/2})$. The same rate of convergence will many times also be true for $\mathrm{E}\,[\,F_n^{\uparrow} - F_n^{\downarrow}\,]$: this will be the case for instance when $F(X) = f(\int_0^1 g(X_s)ds)$ under general assumptions on f, g (e.g. if $|f(y) - f(x)| \le M(x,y)|y - x|$, for a polynomial M, and the same for g; a proof is not essential here). For such a rate (and since the user-specified $F_{n_0}^{\uparrow} - F_{n_0}^{\downarrow}$ should be easily controlled) we will get:

$$\mathrm{P}\,[\,\mathcal{K} > 2^n\,] = \mathcal{O}(2^{-n/2})$$

giving the infinite expectation $\mathrm{E}\,[\,\mathcal{K}\,] = \infty$.

In a given application though, one could fix a big enough maximum number n_{max}, stop the bisections if that number has been reached and report, say, $(F^{\uparrow}_{n_0} + F^{\downarrow}_{n_0})/2$ as the realization of the estimator if that happens, without practical effect on the results. To explain this, note that we know, from (6.5), that the actual unbiased value is either $F^{\uparrow}_{n_0}$ or $F^{\downarrow}_{n_0}$, so we know *precisely* that the absolute bias from the single realization when n_{max} was reached cannot be greater than $(F^{\uparrow}_{n_0} - F^{\downarrow}_{n_0})/2$. In total, when averaging over a number of realizations we can have a precise arithmetic bound on the absolute value of the bias of the reported average; if $n_{\max}$ is 'big enough' so that we reach $n_{\max}$ only in a small proportion of realizations the (analytically known) bias could be of such a magnitude that the reported results will be precisely the same as when implementing the regular algorithm without n_{max} for a reasonably selected degree of accuracy. For example, in the case of the estimation of $\mathrm{E}\,[\,F_c(S)\,]$ in Table 6, we have in fact used $n_{max} = 10$ and found that the introduced bias was smaller than 3×10^{-5} so avoiding it would not make any difference or whatsoever at the results reported right now in Table 6.

Note that such an issue did not arise in the cases of $\mathrm{E}\,[\,F_a(S)\,]$ and $\mathrm{E}\,[\,F_c(S)\,]$ when a reduced version of the ϵ-strong algorithm was applied.

7. Further Directions for Applications

We sketch here some other potential applications of the ϵ-strong algorithm.

In the case of barrier options, sometimes one needs to evaluate expectations involving a Brownian hitting time (see e.g. [19]). Given a non-constant boundary $H : [0, \infty) \to \mathbb{R}$, such that $S_0 < H_0$, consider:

$$\tau_{_H} = \inf\{t \geq 0 : S_t \geq H_t\} \ ,$$

with $S = \{S_t\}$ being the geometric BM in (6.9). The price of a related derivative will be $\mathrm{E}\,[\,F(S)\,]$ where now:

$$F(S) = \psi(S_T) \cdot \mathrm{I}_{\,\tau_{_H} < T}$$

for some pay-off function $\psi(\cdot)$. This estimator requires the evaluation of $\mathrm{I}_{\,\tau_{_H} < T}$ for a realised path. Such an evaluation is possible under our simulation methods, since for a given bridge, say from S_s to S_t with $s < t < T$, we can decide if its maximum is within $[H^{\downarrow}_{s,t}, H^{\uparrow}_{s,t}]$ or not (thus, deciding also whether there is a chance that the bridge hits H on $[s, t]$ or not), with

$$H^{\downarrow}_{s,t} = \inf\{H_u; u \in [s, t]\}, \quad H^{\uparrow}_{s,t} = \sup\{H_u; u \in [s, t]\} \ ,$$

using the refinement procedure described in Section 2.1 (more particularly, a slightly modified version of it, where instead of halving the allowed variation for the maximum, it decides if it lies in a given interval or not). Computational effort will then *only* be spent on the bridges for which the maximum is indeed in $[H^{\downarrow}_{s,t}, H^{\uparrow}_{s,t}]$, iteratively bisecting them until a definite decision is reached about whether H has been hit.

Individual simulation techniques employed in the development of the ϵ-strong algorithm are also of independent interest. For instance, we have exploited during the construction of the ϵ-strong algorithm a monotonic property at the core of the Brownian structure; we can further use this characteristic to develop original simulation techniques for Brownian distributions. One application for instance could involve dynamics of Brownian motion restricted to stay in a bounded domain. A Brownian motion with constant drift, restricted to remain in $(-\pi/2, \pi/2)$, is known (see [16]) to be described via the stochastic differential equation:

$$dX_t = -\tan(X_t)dt + dW_t \ . \tag{7.1}$$

Unbiased sampling methods for X_t are not (to the best of our knowledge) available; one has to resort to Euler, or other, approximations. We can, however, now construct an exact sampling algorithm based on the methods so far described. Girsanov's theorem provides the following expression for the transition density of the Markov process (7.1):

$$\begin{aligned} p(y;x,t) &:= \mathrm{P}\,[\,X_t \in dy \mid X_0 = x\,]\,/\,dy \\ &= \frac{\cos(y)}{\cos(x)}\,\gamma(-\pi/2,\pi/2;t,x,y)\,p_0(y;x,t), \quad y \in (-\pi/2,\pi/2)\ , \end{aligned} \tag{7.2}$$

for the unconditional Brownian transition density:

$$p_0(y;x,t) = (2\pi t)^{-1/2} e^{-(y-x)^2/(2t)} \ .$$

Density (7.2) has a structure reminiscent of that of the density of the middle point in Proposition 5.1: ideas employed there, are also relevant now. Analytically, for x not close to the boundaries, one can simply carry out a rejection sampler with proposals from p_0. Then, the acceptance/rejection decision will be based on comparing a real number with $\gamma(-\pi/2, \pi/2; t, x, y)$ following the pattern described in Section 4.2. As x approaches the boundaries, this algorithm becomes inefficient. But, similarly to the method for the simulation of the density in Proposition 5.1, partial sums from the infinite series-expression for $\gamma(-\pi/2, \pi/2; t, x, y)$ can be incorporated in the proposal to produce an efficient algorithm (Section 5.1 describes the algorithm for the more complex density appearing there; here, we omit the details).

8. Conclusions

We have presented a contribution to sampling methods for Brownian dynamics: a new iterative algorithm that envelopes the Brownian path, thus offering explicit information for all its aspects (minimum, maximum, hitting times). Individual steps of the algorithm could be of independent interest, yielding new sampling methods for distributions derived from Brownian motion dynamics.

We should remark here on the generality of our scope. The ϵ-strong algorithm (or some of its individual steps) can provide, more or less unchanged, unbiased Monte-Carlo estimators in separate estimation problems, for which quite an extensive amount of case-specific methods have been investigated in the literature; for instance, one can refer to the long literature for the applications we briefly described in Sections 6 and 7.

We have presented some applications and sketched some others towards illustrating the potential of our methods. Note that the infinite expectation issue remarked in Section 6.2 is a direct consequence of the Brownian dynamics: the maximum of the Brownian path scales as $\Delta t^{1/2}$ on a small time interval $[0, \Delta t]$ (see e.g. [18]). Thus, any enfolding processes will necessarily converge not faster than $\mathcal{O}(\Delta t^{1/2})$ (which is the order attained by the ϵ-strong algorithm). This relatively slow convergence of the enfolding processes also explains the increased cost for when estimating $\mathrm{E}\,[\,F_b(S)\,]$ in Section 6.1. We envisage that it might be possible to combine the iterative process of the algorithm with a *coupling* step once the bounding processes are relatively close to each other to overcome this long anticipation (in the spirit of [20], [4]). We hope to formalise this idea in future research.

Acknowledgments

We thank the referees for many valuable comments and suggestions that have greatly improved the content of the paper.

References

[1] T. W. Anderson. A modification of the sequential probability ratio test to reduce the sample size. *Ann. Math. Statist.*, 31:165–197, 1960.

[2] L. Bachelier. Théorie de la spéculation. *Ann. Sci. École Norm. Sup. (3)*, 17:21–86, 1900.

[3] Jean Bertoin, Jim Pitman, and Juan Ruiz de Chavez. Constuctions of a Brownian path with a given minimum. *Electron. Comm. Probab.*, 4:31–37 (electronic), 1999.

[4] A. Beskos and G. O. Roberts. One-shop CFTP; application to a class of truncated Gaussian densities. *Methodol. Comput. Appl. Probab.*, 7(4):407–437, 2005.

[5] Alexandros Beskos, Omiros Papaspiliopoulos, and Gareth O. Roberts. Retrospective exact simulation of diffusion sample paths with applications. *Bernoulli*, 12(6):1077–1098, 2006.

[6] Alexandros Beskos, Omiros Papaspiliopoulos, and Gareth O. Roberts. A factorisation of diffusion measure and finite sample path constructions. *Methodol. Comput. Appl. Probab.*, 10(1):85–104, 2008.

[7] Alexandros Beskos, Omiros Papaspiliopoulos, Gareth O. Roberts, and Paul Fearnhead. Exact and computationally efficient likelihood-based estimation for discretely observed diffusion processes. *J. R. Stat. Soc. Ser. B Stat. Methodol.*, 68(3):333–382, 2006. With discussions and a reply by the authors.

[8] Alexandros Beskos and Gareth O. Roberts. Exact simulation of diffusions. *Ann. Appl. Probab.*, 15(4):2422–2444, 2005.

[9] Bruno Casella and Gareth O. Roberts. Exact Monte Carlo simulation of killed diffusions. *Adv. in Appl. Probab.*, 40(1):273–291, 2008.

[10] Luc Devroye. *Nonuniform random variate generation.* Springer-Verlag, New York, 1986.

[11] J. L. Doob. Heuristic approach to the Kolmogorov-Smirnov theorems. *Ann. Math. Statistics*, 20:393–403, 1949.

[12] Paul Glasserman. *Monte Carlo methods in financial engineering*, volume 53 of *Applications of Mathematics (New York).* Springer-Verlag, New York, 2004. Stochastic Modelling and Applied Probability.

[13] Benjamin Jourdain and Mohamed Sbai. Exact retrospective Monte Carlo computation of arithmetic average Asian options. *Monte Carlo Methods Appl.*, 13(2):135–171, 2007.

[14] Ioannis Karatzas and Steven E. Shreve. *Brownian motion and stochastic calculus*, volume 113 of *Graduate Texts in Mathematics.* Springer-Verlag, New York, second edition, 1991.

[15] Bernt Øksendal. *Stochastic differential equations.* Universitext. Springer-Verlag, Berlin, sixth edition, 2003. An introduction with applications.

[16] Ross G. Pinsky. On the convergence of diffusion processes conditioned to remain in a bounded region for large time to limiting positive recurrent diffusion processes. *Ann. Probab.*, 13(2):363–378, 1985.

[17] Klaus Pötzelberger and Liqun Wang. Boundary crossing probability for Brownian motion. *J. Appl. Probab.*, 38(1):152–164, 2001.

[18] Daniel Revuz and Mark Yor. *Continuous martingales and Brownian motion*, volume 293 of *Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences].* Springer-Verlag, Berlin, third edition, 1999.

[19] G. Roberts and C. Shortland. Pricing barrier options with time-dependent coefficients. *Math. Finance*, 7(1):83–93, 1997.

[20] Gareth O. Roberts and Jeffrey S. Rosenthal. One-shot coupling for certain stochastic recursive sequences. *Stochastic Process. Appl.*, 99(2):195–208, 2002.

[21] L. C. G. Rogers and Z. Shi. The value of an Asian option. *J. Appl. Probab.*, 32(4):1077–1088, 1995.